

IMPLEMENTASI ALGORITMA *HILL CLIMBING* DAN ALGORITMA *A** DALAM PENYELESAIAN PENYUSUNAN SUKU KATA DASAR DENGAN POLA PERMAINAN BINTANG KEJORA

Nurdin^[1], Syandriani Harahap^[2]

1. Program Studi Teknik Informatika Universitas Malikussaleh, Aceh

2. Program Studi Teknik Informatika Universitas Malikussaleh, Aceh

E-mail : nurdin_um@ymail.com

Abstrak

Permainan Bintang Kejora merupakan permainan yang sama seperti permainan pergeseran angka dalam kotak berbentuk persegi (*Puzzle*). Jenis permainan seperti ini cenderung mudah untuk diselesaikan. Bentuk wadah bintang menyebabkan arah pergeseran akan menjadi terbatas. Permainan bintang kejora ini cukup rumit dan sukar untuk diselesaikan secara manual. Permainan ini dapat diselesaikan dengan metode heuristik, yaitu dengan menggunakan algoritma *hill climbing* dan algoritma *A**. Sifat algoritma *hill climbing* adalah mencari kemungkinan-kemungkinan dari calon solusi untuk mendapatkan yang optimal bagi penyelesaian masalah dengan mencari nilai heuristik yang terkecil. Sedangkan algoritma *A** membantu menemukan solusi pencarian dalam ruang keadaan dengan mempertimbangkan nilai heuristik terbesar yang dilacak sesuai node yang akan dilewati. Pembuatan perangkat lunak ini dirancang terlebih dahulu dengan menggunakan diagram *State Transition Diagram*, *Use Case Diagram*, *Activity Diagram*, dan *Class Diagram*. Perangkat lunak ini dapat memberikan penyelesaian yang optimum atas permainan bintang kejora yang nantinya menghasilkan huruf yang sebelumnya diacak menjadi tersusun membentuk sebuah kata dasar. Hasil yang didapatkan berupa langkah-langkah ditemukan solusi serta ditampilkan waktu pencarian yang dibutuhkan dalam menemukan solusi. Pencarian *A** lebih cepat menemukan solusi dibandingkan *Hill Climbing*, karena *A** mencari nilai heuristik pada jarak yang terjauh sehingga langsung tepat menuju sasaran.

Kata kunci: *Hill Climbing*, Algoritma *A**, Bintang Kejora

1. PENDAHULUAN

Dalam kehidupan sehari-hari, manusia pada umumnya ingin menyelesaikan permasalahan yang dihadapi dengan secepat-cepatnya dan mendapatkan keuntungan sebanyak-banyaknya dengan mengefesiensikan sumber daya yang dimiliki terhadap batasan-batasan yang ditemui pada suatu masalah. Hal ini juga termasuk dalam menyelesaikan sebuah permainan yang dapat diimplementasikan dalam kehidupan sehari-hari dalam mengasah otak. Keunikan dari permainan-permainan ini menjadikan permainan sangat menyenangkan, dan sekaligus dapat digunakan untuk melatih kecerdasan. Contohnya *Puzzle*, merupakan permainan pergeseran angka yang biasanya dimainkan dalam kotak berbentuk persegi atau persegi panjang. Jenis permainan ini cenderung lebih mudah untuk dimainkan dan diselesaikan. Permainan ini akan menjadi lebih rumit dan sukar apabila dimainkan dalam wadah yang berbentuk bintang. Bentuk wadah ini menyebabkan arah pergeseran akan menjadi terbatas.

Permainan Bintang Kejora ini sebelumnya digunakan untuk memainkan pergeseran angka dari yang teracak sehingga menjadi berurut. Namun disini penulis akan mengganti angka tersebut sehingga menjadi penyusunan suku kata dasar yang teracak agar permainan lebih menyenangkan lagi. Permainan ini dapat diselesaikan dengan menggunakan bantuan struktur pohon pelacakan (*search tree*). Pohon pelacakan adalah suatu pohon (*tree*), dimana akar dari pohon berupa keadaan awal dan cabang merupakan keadaan-keadaan yang mungkin terjadi dari keadaan sebelumnya serta daun merupakan keadaan akhir, yang dapat dijadikan sebagai solusi. Dalam beberapa contoh kasus, ada beberapa atau semua daun bukan merupakan solusi dari permasalahan. Permainan ini dapat diselesaikan dengan berbagai macam Algoritma, diantaranya : Algoritma *Brute Force*, *Greedy*, *Depth First Search*, *Breadth First Search*, *Hill Climbing*, dan *A**. Meskipun Algoritma

penyelesaian banyak, namun tidak semua Algoritma tersebut dapat menjadi solusi terbaik dalam menyelesaikan permainan ini. Didalam penyelesaian permainan Bintang Kejora, dibutuhkan ketepatan, waktu yang singkat dalam mencari solusi dengan rute terpendek namn memiliki optimasi lain. Algoritma pencarian yang akan digunakan adalah Algoritma *Hill Climbing* dan Algoritma *A**.

Algoritma *Hill Climbing* dapat digunakan untuk menyelesaikan kasus ini karena metode hill climbing merupakan variasi dari *dept-first-search* dimana eksplorasi terhadap keputusan dilakukan dengan mencari *path* yang bertujuan menurunkan *cost* untuk menuju kepada *goal*/keputusan melalui nilai heuristik terkecil. Sedangkan Algoritma *A** merupakan algoritma pembandingan yang dapat menguji seberapa baik solusi yang didapat diantara kedua metode ini. Dalam penyelesaian permainan ini, algoritma *A** membantu menemukan solusi pencarian ruang keadaan dengan mempertimbangkan total biaya lintasan yang dilacak sesuai *node* yang akan dilewati. Nilai heuristik yang dipilih adalah nilai heuristik yang paling besar. Dari kedua metode ini diharapkan dapat ditemukan metode mana yang akan menjadi solusi terbaik dalam menyelesaikan permainan Bintang Kejora ini. Permainan bintang kejora ini cukup rumit dan sukar untuk diselesaikan secara manual. Oleh karena itu, penulis berusaha untuk merancang sebuah perangkat lunak yang dapat memberikan solusi bagi permainan ini dengan menggunakan bantuan pohon pelacakan.

2. METODE PENELITIAN

Langkah-langkah yang digunakan dalam penelitian ini adalah sebagai berikut :

1. Pengumpulan Data
Dalam penelitian ini data pokok yang digunakan diperoleh dari Kamus Bahasa Indonesia. Data berupa kosa kata yang terdiri dari 9 huruf.
2. Analisa Data
Dalam tahap ini, yang dilakukan adalah membaca dan mempelajari buku – buku yang berhubungan dengan *Artificial Intelligence* (AI). Mempelajari mengenai proses penyelesaian dari permainan pergeseran angka dalam bintang kejora.
3. Perancangan *Database*
Pada tahap ini penulis membutuhkan sebuah *database* yang berfungsi untuk membantu sistem menyimpan data yang sudah di input. *Database* yang digunakan adalah *Microsoft Access*.
4. Perancangan Sistem
Dalam tahap ini, penulis perlu melakukan sebuah perancangan alur kerja perngkat lunak, juga perancangan antarmuka yang dapat membantu pembuatan aplikasi agar menjadi lebih mudah lagi karena sudah dirancang sebelumnya.
5. Implementasi
Pada tahap ini dilakukan pengkodean untuk mengimplementasikan perancangan sistem yang sudah dibuat sebelumnya kedalam bahasa pemrograman *Visual Basic*
6. Pengujian
Pengujian dilakukan dengan mencoba perangkat lunak yang sudah diselesaika dan memperbaiki kesalahan (*error*) yang muncul.

3. HASIL DAN PEMBAHASAN

3.1 Analisa Sistem

Dalam merancang suatu sistem yang terkomputerisasi, analisa sistem memiliki peranan yang sangat penting dalam membuat rincian aplikasi yang akan dibuat. Analisa sistem bertujuan untuk mengidentifikasi dan mengevaluasi permasalahan yang ada dan nantinya diharapkan dapat menciptakan suatu sistem yang lebih baik.

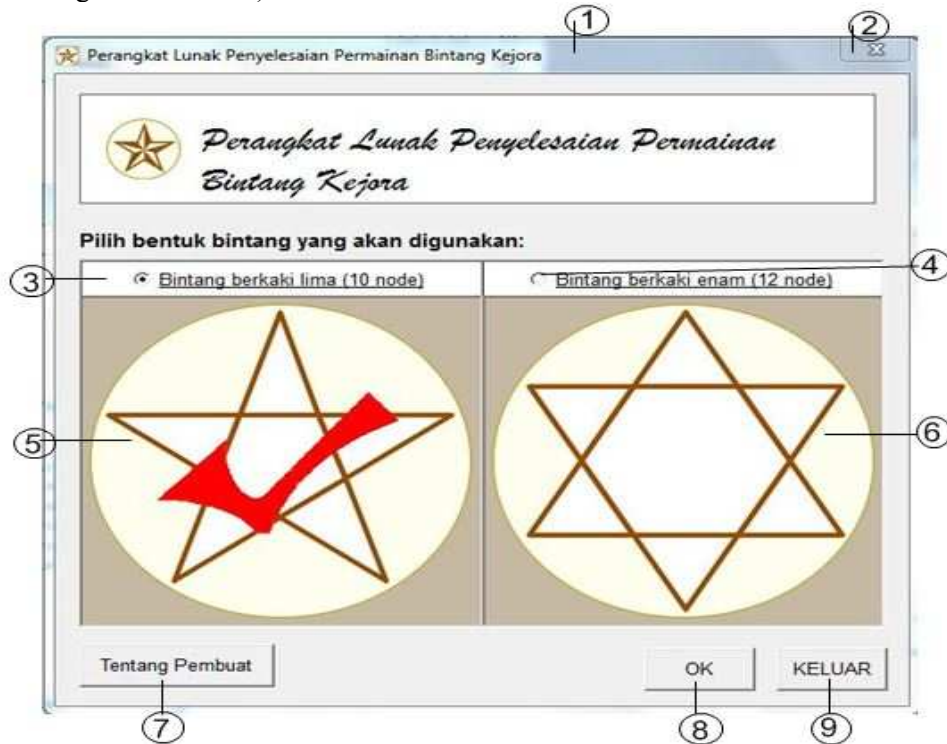
Permainan bintang kejora ini sebelumnya digunakan untuk memainkan pergeseran angka 1 sampai dengan 9 dari yang teracak sehingga menjadi berurut. Bentuk bintang yang digunakan terbagi dua yaitu bintang berkaki lima dan bintang berkaki enam. Untuk bintang berkaki lima, terdapat 10 *node* dengan 9 *node* berisi angka 1 sampai 9 serta sebuah *node* kosong untuk melakukan pergeseran. Sedangkan untuk bintang kejora berkaki enam, terdapat 12 *node* dimana

terdiri 11 *node* yang berisi angka 1 sampai 9 serta ditambahkan huruf A dan B untuk melengkapinya serta sebuah *node* kosong yang digunakan untuk melakukan pergeseran.

3.2 Implementasi Sistem

Formawal

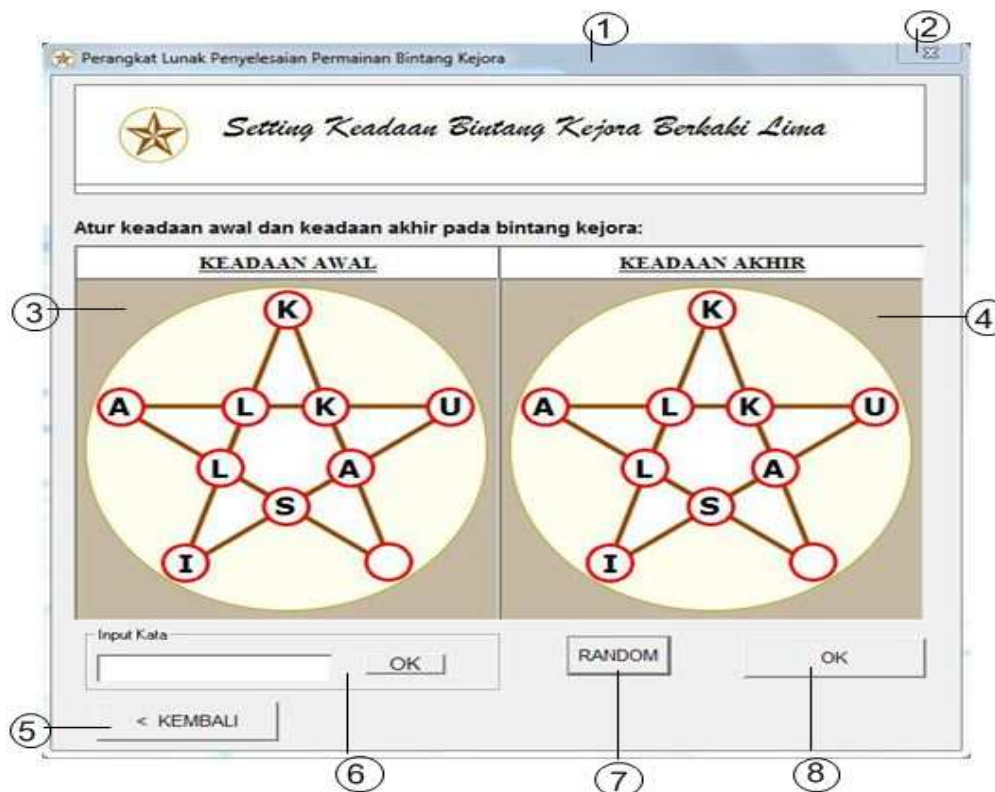
Form ini adalah *form* yang muncul pertama kali ketika program dijalankan. Fungsi dari *form* ini adalah untuk memilih bentuk wadah bintang kejora yang akan digunakan (bintang berkaki lima atau bintang berkaki enam).



Gambar 1. *Form* Awal

Formset keadaan bintang berkaki lima

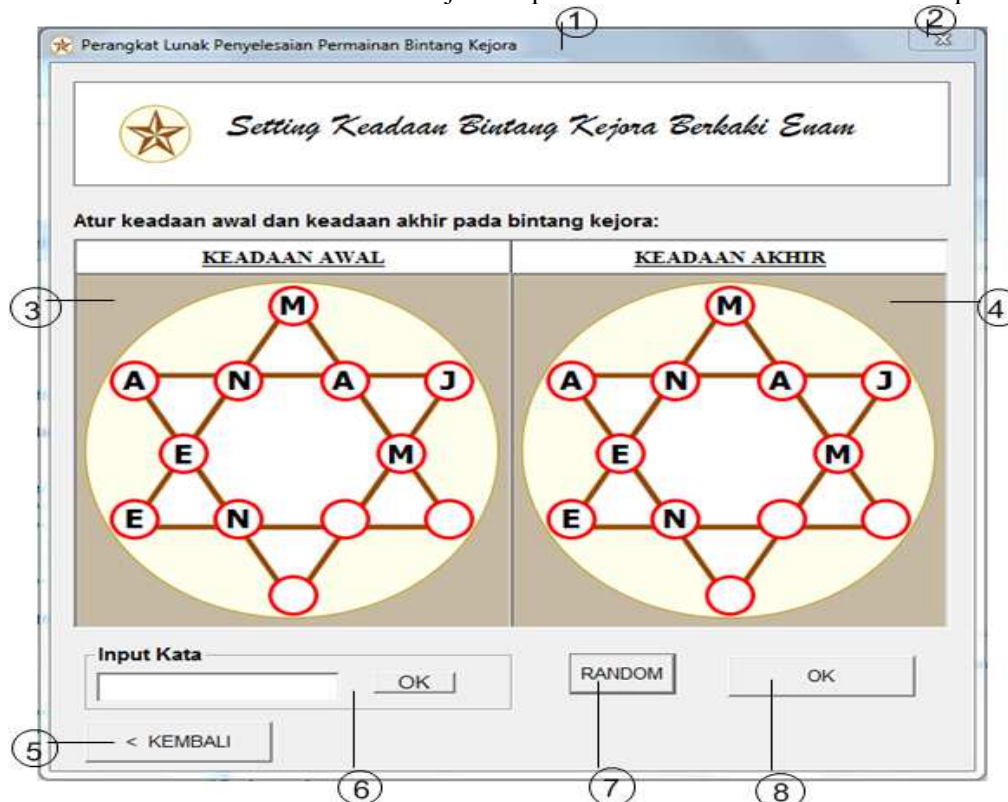
Form ini berfungsi untuk menginput kata dasar dan juga untuk mengatur keadaan awal dan keadaan tujuan pada bintang kejora berkaki lima. Pemilihan kata yang akan dimainkan dapat dihasilkan secara acak oleh komputer bila memilih *button random*, dan dapat diinput secara manual pada *frame input kata*. Namun pengacakan huruf hanya dapat dilakukan secara manual.



Gambar 2. Form Set Keadaan Bintang Berkaki Lima

Formset keadaan bintang berkaki enam

Form ini berfungsi untuk mengatur keadaan awal dan keadaan tujuan pada bintang kejora berkaki enam. Keadaan awal dan keadaan tujuan dapat dihasilkan secara acak oleh komputer.



Gambar 3. Form Set Keadaan Bintang Berkaki Enam

Form pilih metode pencarian

Form ini berfungsi untuk memilih metode pencarian yang akan digunakan untuk mencari solusi pergeseran huruf dari keadaan awal menuju keadaan tujuan.

Metode Pencarian

Pilih salah satu metode pencarian :

☐ **Algoritma Hill Climbing**
Mencari berdasarkan nilai heuristik terkecil. Iterasi akan dilakukan selama nilai heuristik belum mencapai nilai terkecil.

☒ **Algoritma AStar**
Membantu Menemukan solusi dengan mempertimbangkan total biaya lintasan yang akan dilacak sesuai dengan node yang akan dilewati

OK BATAL

Gambar 4. Form Pilih Metode Pencarian

Form input iterasi

Form ini berfungsi untuk melakukan pencarian *hill climbing* apabila ditemukan solusi maka akan menampilkan form solusi.

Input Angka

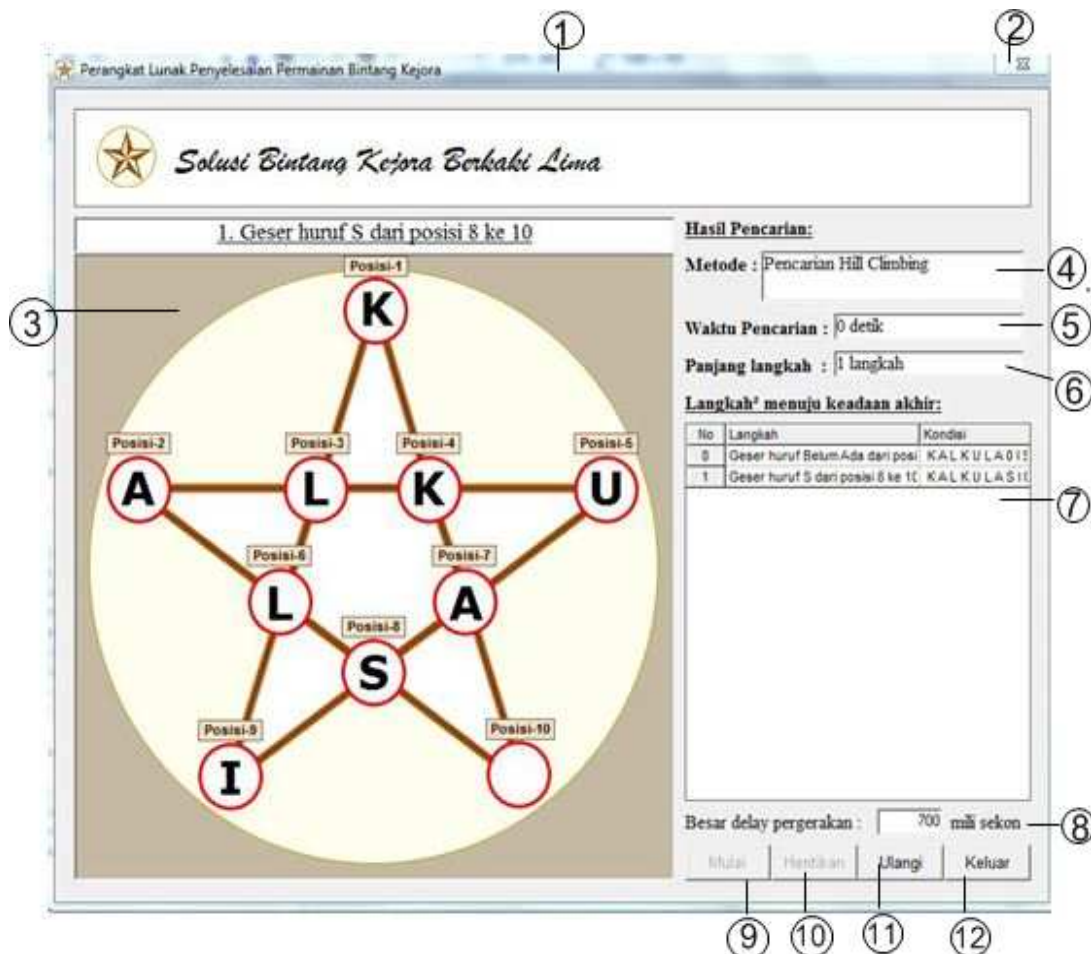
Masukkan Jumlah Iterasi

OK Cancel

Gambar 5. Form Input Iterasi

Form solusi bintang berkaki lima

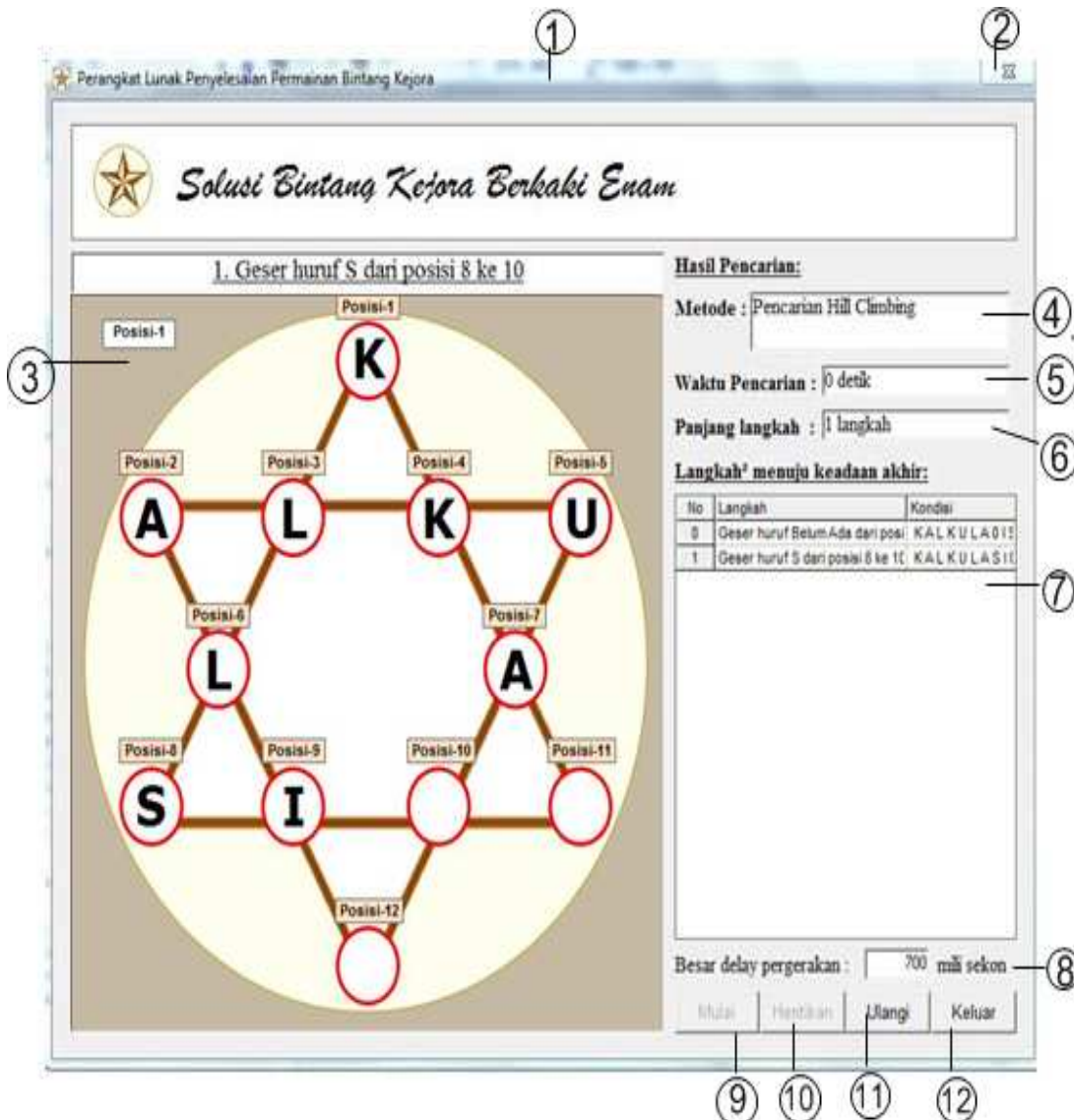
Form ini berfungsi untuk menampilkan solusi pada bintang kejora berkaki lima yang telah didapatkan dari form pencarian.



Gambar 6. Form Solusi Bintang Berkaki Lima

Form solusi bintang berkaki enam

Form ini berfungsi untuk menampilkan solusi pada bintang kejora berkaki enam yang telah didapatkan dari form pencarian.



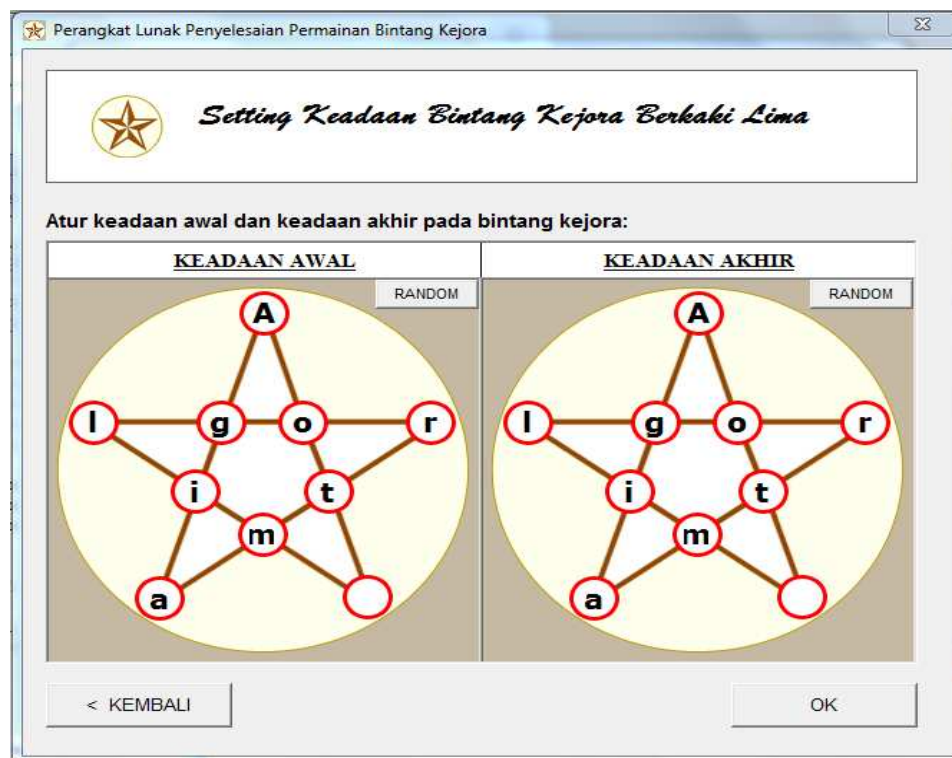
Gambar 7. Form Solusi Bintang Berkaki Enam

Keterangan:

- 1 : Title Bar.
- 2 : tombol 'Close', untuk menutup form.
- 3 : *picturebox*, untuk menampilkan keadaan dari bintang kejora berkaki enam.
- 4 : *label*, untuk menampilkan metode yang telah digunakan untuk mendapatkan solusi.
- 5 : *label*, untuk menampilkan waktu pencarian.
- 6 : *label*, untuk menampilkan panjang langkah solusi.
- 7 : *msflexgrid*, sebagai tabel untuk menampilkan langkah-langkah menuju keadaan akhir.
- 8 : *textbox*, untuk memasukkan besar *delay* pergerakan antar langkah.
- 9 : tombol 'Mulai', untuk memulai proses pergeseran huruf menuju keadaan akhir.
- 10 : tombol 'Hentikan', untuk menghentikan proses pergeseran huruf.
- 11 : tombol 'Ulangi', untuk mengulangi proses pergeseran huruf.
- 12 : tombol 'Keluar', untuk menutup form.

3.3 Pengujian sistem

Untuk menguji solusi yang dihasilkan oleh perangkat lunak, diambil contoh kasus seperti terlihat pada gambar 8. Bentuk bintang yang digunakan adalah bintang kejora berkaki lima.



Gambar 8. Contoh Kasus pada Bintang Kejora Berkaki Lima

Untuk kasus pada gambar 8., solusi yang dihasilkan perangkat lunak dapat dilihat pada tabel 1., dan waktu pencarian yang dibutuhkan dapat dilihat pada tabel 2.

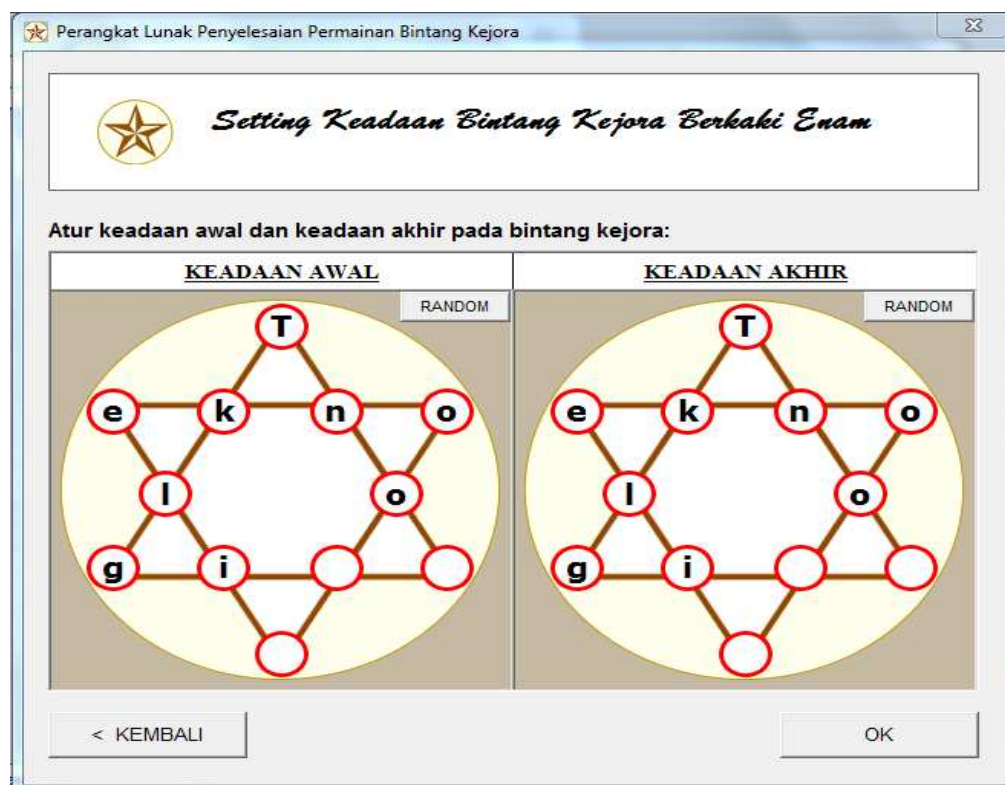
Tabel 1. Solusi Contoh Kasus pada Bintang Kejora Berkaki Lima

No.	Langkah	Kondisi
0.	Keadaan Awal.	0LAGRIOTAM
1.	Geser huruf 'A' dari posisi 3 ke posisi 1.	AL0GRIOTAM
2.	Geser huruf 'G' dari posisi 4 ke posisi 3.	ALG0RIOTAM
3.	Geser huruf 'O' dari posisi 7 ke posisi 4.	ALGORI0TAM
4.	Geser huruf 'T' dari posisi 8 ke posisi 7.	ALGORIT0AM
5.	Geser huruf 'M' dari posisi 10 ke posisi 8.	ALGORITMA0

Tabel 2. Waktu Pencarian Solusi pada Bintang Kejora Berkaki Lima

No.	Metode Pencarian	Waktu Pencarian
1.	Pencarian Hill Climbing.	0.0019375 detik
2.	Pencarian A*.	0.00003125 detik

Untuk contoh kasus pada gambar 8., solusi yang dihasilkan oleh pencarian Hill Climbing dan A* adalah sama (lihat tabel 1.). Perbedaannya hanya terletak pada waktu pencarian (lihat tabel 2.). Pada contoh kasus tersebut, waktu pencarian dengan pencarian *hill climbing* biasa adalah 0.0019375 detik, sedangkan pencarian A* membutuhkan waktu 0.00003125 detik untuk menemukan solusi. Pencarian A* membutuhkan waktu yang lebih cepat dalam menemukan solusi dibandingkan dengan pencarian *hill climbing*. Hal ini dikarenakan pencarian A* memprioritaskan huruf yang memiliki jarak yang terjauh, sehingga dapat langsung menempati posisi yang tepat. Sedangkan pada pencarian *Hill Climbing*, terdapat resiko dimana solusi tidak bisa ditemukan karena hanya menghitung berdasarkan jumlah heuristik terkecil, dimana yang memiliki jumlah salah terkecil saja yang digunakan langkahnya. Contoh kasus pada bintang kejora berkaki enam seperti terlihat pada gambar 9., dibawah ini.



Gambar 9. Contoh Kasus pada Bintang Kejora Berkaki Enam

Untuk kasus pada gambar 9., solusi yang dihasilkan perangkat lunak dapat dilihat pada tabel 3., dan waktu pencarian yang dibutuhkan dapat dilihat pada tabel 4.

Tabel 3. Solusi Contoh Kasus pada Bintang Kejora Berkaki Enam

No.	Langkah	Kondisi
0.	Keadaan Awal.	00ETOKNLGO0I
1.	Geser huruf 'T' dari posisi 4 ke posisi 1.	T0E0OKNLGO0I
2.	Geser huruf 'E' dari posisi 3 ke posisi 2.	TE00OKNLGO0I
3.	Geser huruf 'N' dari posisi 7 ke posisi 4.	TE0NOK0LGO0I
4.	Geser huruf 'K' dari posisi 6 ke posisi 3.	TEKNO00LGO0I
5.	Geser huruf 'L' dari posisi 8 ke posisi 6.	TEKNOL00GO0I

6.	Geser huruf 'O' dari posisi 10 ke posisi 7.	TEKNOLO0G00I
7.	Geser huruf 'G' dari posisi 9 ke posisi 8.	TEKNOLOG000I
8.	Geser huruf 'I' dari posisi 12 ke posisi 9.	TEKNOLOGI000

Tabel 4. Waktu Pencarian Solusi pada Bintang Kejora Berkaki Enam

No.	Metode Pencarian	Waktu Pencarian
1.	Pencarian Hill Climbing	0.00034375 detik
2.	Pencarian A*.	Solusi tidak ditemukan

Apabila dilakukan beberapa kali pencarian *Hill Climbing* pada contoh kasus pada gambar 9., maka waktu pencarian yang dihasilkan dapat dilihat pada table 5., berikut.

Tabel5. Waktu Pencarian pada Pencarian *Hill Climbing*

Pencarian	Waktu Pencarian
Pencarian ke 1.	0.000625 detik
Pencarian ke 2.	0.000718 detik
Pencarian ke 3.	0.0005 detik
Pencarian ke 4.	0.001656 detik
Pencarian ke 5.	0.00175 detik

Pada tabel 5., terlihat bahwa waktu pencarian yang dibutuhkan berbeda-beda. Hal ini disebabkan komputasi memori komputer untuk melakukan pekerjaan (mencari solusi) tidak selalu sama persis pada beberapa waktu yang berbeda.

4. KESIMPULAN

1. Pencarian *Hill Climbing* lebih efektif dalam menemukan solusi daripada pencarian *A**.
2. Pencarian *A** membutuhkan waktu sedikit lebih cepat dalam menemukan solusi dibandingkan dengan pencarian *Hill Climbing*.
3. Pada bentuk bintang berkaki enam, pencarian *A** hanya dapat menemukan solusi pada beberapa kata dengan pergeseran beberapa langkah saja. Hal ini disebabkan karena tidak semua langkah yang memiliki jarak yang jauh adalah solusi yang terbaik sehingga banyak kata yang nilainya tidak bisa ditemukan. Dapat disimpulkan bahwa pencarian *A** tidak mendukung untuk menemukan solusi dalam pola bintang berkaki enam.
4. Pada pencarian *Hill Climbing*, terdapat resiko dimana solusi tidak bisa ditemukan karena hanya menghitung berdasarkan jumlah heuristik terkecil, dimana yang memiliki jumlah salah terkecil saja yang digunakan langkahnya.
5. Perangkat lunak ini merupakan implementasi nyata penggunaan metode pencarian *Hill Climbing* dan pencarian *A** dalam mencari solusi dalam suatu permasalahan berbasis *Artificial Intelligence* (AI).

5. SARAN

1. Perangkat lunak dapat dikembangkan dengan menambahkan beberapa bentuk wadah lainnya, seperti: segi enam (*heksagon*), segi tujuh (*heptagon*), segi delapan (*octagon*) dan bentuk lainnya.
2. Perangkat lunak ini dapat dikembangkan dengan menambahkan metode pencarian lainnya, seperti: metode pencarian *Depth-First-Search*, *Pencarian Heuristik*, dan metode lainnya.
3. Perangkat lunak dapat dikembangkan dengan menambahkan penggambaran dan penjelasan pada pohon pelacakan yang digunakan untuk mencari solusi.

DAFTAR PUSTAKA

- Anggreini Susanti, 2014. *Kecerdasan Buatan: Simple Hill Climbing*. Fakultas Sains dan Teknologi, Jurusan Matematika, Universitas Airlangga.
- Azizah Zakiah, 2012. *Penyelesaian Masalah 8 Puzzle Dengan Algoritma Hill Stepest Ascent Loglist Heuristik Berbasis Java*. Fakultas teknik, Jurusan Teknik Informatika, Politeknik Pos Indonesia.
- Chandra Usman, Hendri Sopryadi, 2013. *Rancang Bangun Game Slider Puzzle Berbasis Android Menggunakan Metode Heuristik dengan Teknik Best First Search*. STMIK MDP.
- Feddy Setio, Anggraini Mulwinda, 2010. *Pencarian Rute Terpendek dengan Menggunakan Algoritma Depth First, Breath First dan Hill Climbing*. Jurusan Teknik Elektro, Universitas Negeri Semarang.
- Haviluddin, 2011. *Memahami Penggunaan UML (Unified Modeling Language)*. Program Studi Ilmu Komputer, FMIPA, Universitas Mulawarman.
- Hengky Alexander Mangkulo, 2011. *Cara Mudah Mengenal Visual Basic 6.0*. Penerbit Elex Media Komputindo.
- Latius Hermawan, R. Kristoforus, 2013. *Penerapan Algoritma A* Pada Aplikasi Puzzle*. Fakultas Teknik, Jurusan Teknik Informatika, Sekolah Tinggi Teknik Musi.
- Muhammad Sadeli, 2010. *Kumpulan Proyek Visual Basic 6.0*. Penerbit Maxikom, Palembang.
- Muh. Yamin, Moh. Bandrigo, 2015. *Aplikasi Pencarian Jalur Terpendek Menggunakan Algoritma A**. Jurusan Teknik Informatika, FTEKNIK YHO, Kendari.
- Ruri Hartika Zain, 2012. *Representasi Pengetahuan Berbasis Rule dalam Menganalisa Kekurangan Vitamin Pada Tubuh Manusia*. Jurnal Teknologi Informasi dan Pendidikan.
- Widodo Budihartono, Derwin Suhartono, 2014. *Artificial Intelligence – Konsep dan Penerapannya*. Penerbit ANDI, Yogyakarta.